Government Smart Card Interoperability Specification v2.1 (NISTIR 6887, 2003 Edition) Virtual Card Edge Interface Virtual Machine Cards

Conformance Test Assertions

DRAFT

Alan Goldfine

April 5, 2004

This document contains the conformance test assertions for each of the APDUs comprising the Virtual Machine Virtual Card Edge Interface (VCEI) of version 2.1 of the Government Smart Card Interoperability Specification (GSC-IS), as contained in NIST Interagency Report 6887, 2003 Edition.

The 12 sections of this document correspond to the 12 APDUs in the VCEI, as specified in Section 5.3. The test assertions for each of the APDUs are numbered in the form X.Y, where X is the APDU number, and Y is the number of the assertion for that APDU. Thus, 4.7 is the number given to assertion 7 for APDU 4 (GET ACR).

1. SELECT APPLET

CLA	00
INS	A4
P1	04
P2	00
L_c	Length of the applet AID (between 5 and 16 bytes)
Data Field	Applet AID
L_e	Empty

References:

1. GSC-IS 5.3.3.2.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 1.1

Purpose: To test SELECT APPLET using valid parameters.

Scenario:

1. L_c == the length of the applet AID in the data field

2. Data Field == the AID of an applet on the card.

3. A SELECT APPLET APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00.

2. The specified applet is selected.

Assertion 1.2

Purpose: To test SELECT APPLET in the case where the applet is logically deleted. *Anybody know what this means? How do you logically delete an applet?*

Scenario:

1. L_c == the length of the applet AID in the data field

2. Data Field == the AID of a logically deleted applet on the card.

3. A SELECT APPLET APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 62 00.

Assertion 1.3

Purpose: To test SELECT APPLET using an invalid L_c.

Scenario:

1. L_c /= the length of the applet AID in the data field

2. Data Field == the AID of an applet on the card.

3. A SELECT APPLET APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 1.4

Purpose: To test SELECT APPLET where the specified applet is not on the card.

Scenario:

1. L_c == the length of the applet AID in the data field

2. Data Field /= the AID of any applet on the card.

3. A SELECT APPLET APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 99, 6A 80, or 6A 82.

Assertion 1.5

Purpose: To test SELECT APPLET using invalid parameters P1-P2

Scenario:

1. At least one of the following is true:
   - P1 /= 04
   - P2 /= 00.

2. L_c == the length of the applet AID in the data field

3. Data Field == the AID of an applet on the card.

4. A SELECT APPLET APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86.

2. SELECT OBJECT

CLA	00
INS	A4
P1	02
P2	00
L_c	Length of the object ID (2 bytes)
Data Field	Object ID
L_e	Empty

References:

1. GSC-IS 5.3.3.3.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.
2. An applet is currently selected.

Assertion 2.1

Purpose: To test SELECT OBJECT using valid parameters.

Scenario:

1. L_c == the length of the object ID in the data field (2 bytes)
2. Data Field == the ID of an object managed by the current applet.
3. A SELECT OBJECT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00.
2. The specified object is selected.

Assertion 2.2

Purpose: To test SELECT OBJECT using an invalid L_c.

Scenario:

1. L_c /= the length of the object ID in the data field
2. Data Field == the ID of an object managed by the current applet.
3. A SELECT OBJECT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 2.3

Purpose: To test SELECT OBJECT where the specified object is not managed by the current applet.

Scenario:

1. L_c == the length of the object ID in the data field

2. Data Field /= the ID of any object managed by the current applet.

3. A SELECT OBJECT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 80 or 6A 82.

Assertion 2.4

Purpose: To test SELECT OBJECT using invalid parameters P1-P2

Scenario:

1. At least one of the following is true:
   - P1 /= 04
   - P2 /= 02.

2. L_c == the length of the object ID in the data field.

3. Data Field == the ID of an object managed by the current applet.

4. A SELECT OBJECT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86.

3. GET PROPERTIES

CLA	00
INS	56
P1	00 Get GSC-IS v2.0 compatible properties *(Note: No assertions are provided, at this time, for the case P1==00)* 01 Get all properties 02 Get properties of the tags specified in the data field
P2	00
L_c	If P2==02, then length of list of specified tags If P2/=02, then empty
Data Field	If P2==02, then a list of specified tags If P2/=02, then empty
L_e	A value that is <= the length of the list of the TLVs requested by P1

References:

1. GSC-IS 5.3.3.4.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. An applet is currently selected.

Assertion 3.1

Purpose: To test GET PROPERTIES using valid parameters (all properties).

Scenario:

1. P1 == 01.

2. L_c == empty.

3. Data Field == empty.

4. L_e == Expected length of the list of TLVs to be returned.

5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if L_e == the length of the total list of TLVs

   or
   - SW1 SW2 == 61 LL, if L_e < the length of the total list of TLVs. LL is the size of the next block of the TLV list that is available to be read..

2. Response Data Field == the first L_e bytes of the total list of TLVs associated with the selected applet.

Assertion 3.2

Purpose: To test GET PROPERTIES using valid parameters (list of tags).

Scenario:

1. P1 == 02.
2. L_c == length of list of specified tags in data field.
3. Data Field == list of specified tags.
4. L_e == Expected length of the list of TLVs to be returned.
5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 90 00, if L_e == the length of the list of TLVs requested to be returned

   or
    - SW1 SW2 == 61 LL, if L_e < the length of the list of TLVs requested to be returned. LL is the size of the next block of the TLV list that is available to be read.
2. Response Data Field == the first L_e bytes of the list of TLVs requested to be returned.

Assertion 3.3

Purpose: To test GET PROPERTIES using an incorrect parameter L_c.

Scenario:

1. P1 == 02.
2. L_c /= length of list of specified tags in data field.
3. Data Field == list of specified tags.
4. L_e == Expected length of the list of TLVs to be returned.
5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 67 00.

Assertion 3.4

Purpose: To test GET PROPERTIES using the wrong length in the L_e parameter (all properties).

Scenario:

1. P1 == 01.

2. L_c == empty.

3. Data Field == empty.

4. L_e > the length of the total list of TLVs.

5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6C XX.

Assertion 3.5

Purpose: To test GET PROPERTIES using invalid parameters in the Data Field.

Scenario:

1. P1 == 02.

2. L_c == length of list in data field.

3. Data Field == an invalid list of tags.

4. L_e == Expected length of the list of TLVs to be returned.

5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 80.

Assertion 3.6

Purpose: To test GET PROPERTIES using invalid parameters P1-P2 (all properties).

Scenario:

1. At least one of the following is true:
   - P1 /= 00, 01, or 02
   - P2 /= 00.

2. L_c == empty.

3. Data Field == empty.

4. L_e == Expected length of the list of TLVs to be returned.

5. A GET PROPERTIES APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86 or 6A 88.

4. GET ACR

CLA	80
INS	4C
P1	00 Get complete ACR table
	01 Get one ACR table entry based on ACRID
	10 Get complete Applet/Object ACR table
	11 Get Applet/Object ACR table entries for one applet based on applet AID
	12 Get one entry of the Applet/Object ACR table based on object ID
	20 Get complete Access Method Provider table
	21 Get complete Service Applet table
P2	00
L_c	If P2==00, then empty
	If P2==01, then the length of the ACRID in the data field (01)
	If P2==10, then empty
	If P2==11, then the length of the applet AID in the data field
	If P2==12, then the length of the object ID in the data field (02)
	If P2==20, then empty
	If P2==21, then empty
Data Field	If P2==00, then empty
	If P2==01, then the ACRID
	If P2==10, then empty
	If P2==11, then the applet AID
	If P2==12, then the object ID
	If P2==20, then empty
	If P2==21, then empty
L_e	empty

References:
1. GSC-IS 5.3.3.5.

Starting State for Each Assertion:
1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. An applet is currently selected.

Assertion 4.1

Purpose: To test GET ACR using valid parameters (complete ACR table).

Scenario:
1. P1 == 00.

2. L_c == empty.

3. Data Field == empty.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of ACE table data that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of ACR table data.

Assertion 4.2

Purpose: To test GET ACR using valid parameters (single ACR table entry based on ACRID).

Scenario:

1. P1 == 01.

2. L_c == the length of the ACRID in the Data Field (01).

3. Data Field == the ACRID.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the ACR table entry data that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of the ACR table entry data.

Assertion 4.3

Purpose: To test GET ACR using valid parameters (complete applet/object table).

Scenario:

1. P1 == 10.

2. L_c == empty.

3. Data Field == empty.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the applet/object ACR table entry data that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of the applet/object ACR table entry data.

Assertion 4.4

Purpose: To test GET ACR using valid parameters (Applet/Object ACR table entries for one applet based on applet AID).

Scenario:

1. P1 == 11.

2. L_c == the length of the applet AID in the Data Field.

3. Data Field == the AID of an applet on the card.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the Applet/Object ACR table entries data for the specified applet that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of the Applet/Object ACR table entries data.

Assertion 4.5

Purpose: To test GET ACR using valid parameters (one entry of the Applet/Object ACR table based on object ID).

Scenario:

1. P1 == 12.

2. L_c == the length of the object AID in the Data Field (02).

3. Data Field == the ID of an object on the card.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the Applet/Object ACR table entries data for the specified object that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of the Applet/Object ACR table entries data.

Assertion 4.6

Purpose: To test GET ACR using valid parameters (complete Access Method Provider table).

Scenario:

1. P1 == 20.

2. L_c == empty.

3. Data Field == empty.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the Access Method Provider table data that is available to be read.

2. Response Data Field == the Applet Information String, followed by the first block of the Access Method Provider table data.

Assertion 4.7

Purpose: To test GET ACR using valid parameters (complete Service Applet table).

What are Service Applets? Are they the applets that manage datan objects? Is the applet that manages the CCC a Service applet? What about the applet that manages the Master File?

Scenario:

1. P1 == 21.
2. L_c == empty.
3. Data Field == empty.
4. L_e == empty.
5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00, if the length of the total data to be returned is <256

   or
   - SW1 SW2 == 61 LL, if the length of the total data to be returned is >=256. LL is the length of the next block of the Service Applet table data that is available to be read.
2. Response Data Field == the Applet Information String, followed by the first block of the Service Applet table data.

Assertion 4.8

Purpose: To test GET ACR in the case where the specified applet is logically deleted. *Anybody know what this means? How do you logically delete an applet?* (Applet/Object ACR table entries for one applet based on applet AID).

Scenario:

1. P1 == 11.
2. L_c == the length of the applet AID in the Data Field.
3. Data Field == the AID of a logically deleted applet on the card.
4. L_e == empty.
5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 62 00.

Assertion 4.9

Purpose: To test GET ACR using an incorrect parameter L_c (single ACR table entry based on ACRID).

Scenario:

1. P1 == 01.

2. L_c /= the length of the ACRID in the Data Field (01).

3. Data Field == the ACRID.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 4.10

Purpose: To test GET ACR where the specified applet is not on the card (Applet/Object ACR table entries for one applet based on applet AID).

Scenario:

1. P1 == 11.

2. L_c == the length of the applet AID in the Data Field.

3. Data Field /= the AID of an applet on the card.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 99, or 6A 80, or 6A 82.

Assertion 4.11

Purpose: To test GET ACR using an incorrect P1 or P2 parameter.

Scenario:

1. At least one of the following is true:
   - P1 /= 00, 01, 10, 11, 12, 20, or 21
   - P2 /= 00.

2. L_c == empty.

3. Data Field == empty.

4. L_e == empty.

5. A GET ACR APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86.

5. GET RESPONSE

CLA	00
INS	C0
P1	00
P2	00
L_c	empty
Data Field	empty
L_e	number of bytes to read in response

References:

1. GSC-IS 5.3.3.6.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The immediately preceding APDU has indicated that a block of L bytes of additional data is available to be read.

Assertion 5.1

Purpose: To test GET RESPONSE using valid parameters, with the number of bytes specified to be retrieved equal to the maximum available.

Scenario:

1. L_e == L.

2. A GET RESPONSE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00
   - Response Data Field == the string of bytes, of length L_e, that is read.

Assertion 5.2

Purpose: To test GET RESPONSE using valid parameters, where the number of bytes specified to be retrieved is less than the maximum available.

Scenario:

1. L_e < L.

2. A GET RESPONSE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 61 XX, where XX is L - L_e
   - Response Data Field == the string of bytes, of length L_e, that is read.

Assertion 5.3

Purpose: To test GET RESPONSE where the number of bytes specified to be retrieved is greater than the maximum available.

Scenario:

1. A GET RESPONSE APDU is issued, using
   - L_e > L.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6C XX.

Assertion 5.4

Purpose: To test GET RESPONSE using an invalid P1 or P2 parameter.

Scenario:

1. L_e == L.

2. At least one of P1 or P2 is /= 0.

3. A GET RESPONSE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86.

6. VERIFY PIN

CLA	00
INS	20
P1	00
P2	00
L_c	00, or NN if a PIN is specified in Data Field
Data Field	empty or PIN code to be verified
L_e	empty

References:
1. GSC-IS 5.3.3.7.

Starting State for Each Assertion:
1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 6.1

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN not yet verified, no PIN code specified).

Scenario:
1. A PIN code is required for the services of the currently selected object.

2. The PIN code has not already been verified.

3. X == the number of allowable PIN tries.

4. L_c == 00.

5. Data Field == empty.

6. A VERIFY PIN APDU is issued.

Expected Results:
1. The APDU returns
   - SW1 SW2 == 63 C(X-1) (PIN not verified).

2. The services of the currently selected object cannot be performed.

Assertion 6.2

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN not yet verified, correct PIN code specified).

Scenario:
1. A PIN code is required for the services of the currently selected object.

2. The PIN code has not already been verified.

3. L_c == the length of the PIN in the data field.

4. Data Field == the correct PIN code for the services of the currently selected object.

5. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The services of the currently selected object can be performed.

Assertion 6.3

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN not yet verified, incorrect PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.

2. The PIN code has not already been verified.

3. X == the number of allowable PIN tries.

4. L_c == the length of the PIN in the data field.

5. Data Field == an incorrect PIN code for the services of the currently selected object.

6. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 63 C(X-1) (PIN not verified).

2. The services of the currently selected object cannot be performed.

Assertion 6.4

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN has been verified, no PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.

2. The PIN code has already been verified.

3. L_c == 00.

4. Data Field == empty.

5. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The services of the currently selected object can be performed.

Assertion 6.5

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN has been verified, correct PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.

2. The PIN code has already been verified.

3. L_c == the length of the PIN in the data field.

4. Data Field == the correct PIN code for the services of the currently selected object.

5. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The services of the currently selected object can be performed.

Assertion 6.6

Purpose: To test VERIFY PIN using valid parameters (PIN verification required, PIN has been verified, incorrect PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.

2. The PIN code has not already been verified.

3. L_c == the length of the PIN in the data field.

4. Data Field == an incorrect PIN code for the services of the currently selected object.

5. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 (Successful execution).

2. The services of the currently selected object can be performed.

Assertion 6.7

Purpose: To test VERIFY PIN using valid parameters (PIN verification not required, no PIN code specified).

Scenario:

1. A PIN code is not required for the services of the currently selected object.
2. L_c == 00.
3. Data Field == empty.
4. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 6A 88 (No PIN code defined).
2. The services of the currently selected object can be performed.

Assertion 6.8

Purpose: To test VERIFY PIN using valid parameters (PIN verification not required, PIN code specified).

Scenario:

1. A PIN code is not required for the services of the currently selected object.
2. L_c == the length of the PIN in the data field.
3. Data Field == a PIN.
4. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 6A 88 (No PIN code defined).
2. The services of the currently selected object can be performed.

Assertion 6.9

Purpose: To test VERIFY PIN using a blocked PIN (correct PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.
2. The PIN code has not already been verified.

3. The PIN is currently blocked.

4. L_c == the length of the PIN in the data field.

5. Data Field == the correct PIN code for the services of the currently selected object.

6. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 83 (PIN code blocked).

2. The services of the currently selected object cannot be performed.

Assertion 6.10

Purpose: To test VERIFY PIN using an invalid L_c (PIN code specified).

Scenario:

1. A PIN code is required for the services of the currently selected object.

2. The PIN code has not already been verified.

3. L_c /= the length of the PIN in the data field.

1. Data Field == the correct PIN code for the services of the currently selected object.

2. A VERIFY PIN APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

7. READ BUFFER

CLA	80
INS	52
P1	MSB of offset in buffer from which data is to be read
P2	LSB of offset in buffer from which data is to be read
L_c	02
Data Field	01•number of bytes to read from buffer, for T-buffer 02•number of bytes to read from buffer, for V-buffer
L_e	empty

References:

1. GSC-IS 5.3.4.2.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 7.1

Purpose: To test READ BUFFER using valid parameters (T-buffer).

Scenario:

1. An object is currently selected. The ACR for the read service for this object has been satisfied.

2. P1P2 represents an offset that is within the bounds of the T-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the T-buffer of the currently selected object.

4. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 and Response Data Field == the string of bytes specified to be read

   or
   - 61 XX with XX bytes remaining to be read.

Assertion 7.2

Purpose: To test READ BUFFER using valid parameters (V-buffer).

Scenario:

1. An object is currently selected. The ACR for the read service for this object has been satisfied.

2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.

4. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00 and Response Data Field == the string of bytes specified to be read

   or
   - 61 XX with XX bytes remaining to be read.

Assertion 7.3

Purpose: To test READ BUFFER using an incorrect parameter L_c (V-buffer).

Scenario:

1. An object is currently selected. The ACR for the read service for this object has been satisfied.

2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.

4. L_c /= 02.

5. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 7.4

Purpose: To test READ BUFFER using an invalid parameter in the Data Field.

Scenario:

1. An object is currently selected. The ACR for the read service for this object has been satisfied.

2. P1P2 represents an offset that is within the bounds of both the V-buffer and the T-buffer of the currently selected object.

3. The first byte of the Data Field /= 01 or 02.

4. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 80 or 6A 88.

Assertion 7.5

Purpose: To test READ BUFFER using an invalid P1 or P2 parameter (V-buffer).

Scenario:

1. An object is currently selected. The ACR for the read service for this object has been satisfied.
2. P1P2 represents an offset that is outside the bounds of the V-buffer of the currently selected object.
3. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 80.

Assertion 7.6

Purpose: To test READ BUFFER where the security status of the read service of the currently selected object is not satisfied (V-buffer).

Scenario:

1. An object is currently selected. The ACR for the read service for this object has not been satisfied.
2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.
3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.
4. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 82.

Assertion 7.7

Purpose: To test READ BUFFER when no container is currently selected.

Scenario:

1. No object is currently selected.
2. A READ BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 82.

8. UPDATE BUFFER

CLA	80
INS	58
P1	MSB of offset in buffer into which data is to be written
P2	LSB of offset in buffer into which data is to be written
L_c	01 + length of data to be updated
Data Field	01•the update data, for the T-buffer
	02•the update data, for the V-buffer
L_e	empty

References:
1. GSC-IS 5.3.4.1.

Starting State for Each Assertion:
1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 8.1

Purpose: To test UPDATE BUFFER using valid parameters (T-buffer).

Scenario:
1. An object is currently selected. The ACRs for the update and read services for this object have been satisfied.

2. P1P2 represents an offset that is within the bounds of the T-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the T-buffer of the currently selected object.

4. An UPDATE BUFFER APDU is issued.

Expected Results:
1. The APDU returns
   - SW1 SW2 == 90 00.

Assertion 8.2

Purpose: To test UPDATE BUFFER using valid parameters (V-buffer).

Scenario:
1. An object is currently selected. The ACRs for the update and read services for this object have been satisfied.

2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.

4. An UPDATE BUFFER APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 90 00.

Assertion 8.3

Purpose: To test UPDATE BUFFER using an incorrect parameter L_c (V-buffer).

Scenario:

1. An object is currently selected. The ACRs for the update and read services for this object have been satisfied.

2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.

4. L_c /= 1 + length of update data.

5. An UPDATE BUFFER APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 67 00.

Assertion 8.4

Purpose: To test UPDATE BUFFER using an invalid parameter in the data field.

Scenario:

1. An object is currently selected. The ACRs for the update and read services for this object have been satisfied.

2. P1P2 represents an offset that is within the bounds of both the V-buffer and the T-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of both the V-buffer and the T-buffer of the currently selected object.

4. The first byte of the Data Field /= 01 or 02.

5. An UPDATE BUFFER APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 6A 80 or 6A 88.

Assertion 8.5

Purpose: To test UPDATE BUFFER using an invalid P1 or P2 parameter (V-buffer).

Scenario:

1. An object is currently selected. The ACRs for the update and read services for this object have been satisfied.

2. P1P2 represents an offset that is outside the bounds of the V-buffer of the currently selected object.

3. An UPDATE BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86.

Assertion 8.6

Purpose: To test UPDATE BUFFER where the security status of the currently selected object is not satisfied (V-buffer).

Scenario:

1. An object is currently selected. The ACR for the update service for this object has not been satisfied. The ACR for the read service for this object has been satisfied.

2. P1P2 represents an offset that is within the bounds of the V-buffer of the currently selected object.

3. P1P2 + the second byte of the Data Field is within the bounds of the V-buffer of the currently selected object.

4. An UPDATE BUFFER APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 82.

Assertion 8.7

Purpose: To test UPDATE BUFFER when no container is currently selected.

Scenario:

1. No object is currently selected.

2. An UPDATE BUFFER APDU is issued.

Expected Results:

2. The APDU returns
   - SW1 SW2 == 6A 82.

9. GET CHALLENGE

CLA	00
INS	84
P1	00
P2	00
L_c	empty
Data Field	empty
L_e	challenge length (08)

References:
1. GSC-IS 5.3.5.1.

Starting State for Each Assertion:
1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 9.1

Purpose: To test GET CHALLENGE using valid parameters.

Scenario:
1. A GET CHALLENGE APDU is issued.

Expected Results:
1. The APDU returns
   - SW1 SW2 == 90 00
   - Response Data Field == L_e bytes representing the cryptographic challenge.

Assertion 9.2

Purpose: To test GET CHALLENGE where the specified length of the returned challenge is incorrect.

Scenario:
1. L_e /= 08.

2. A GET CHALLENGE APDU is issued.

Expected Results:
1. The APDU returns
   - SW1 SW2 == 6C 08.

Assertion 9.3

Purpose: To test GET CHALLENGE using an invalid P1 or P2 parameter.

Scenario:
1. At least one of P1 or P2 is /= 0.

2. A GET CHALLENGE APDU is issued.

Expected Results:

1. The APDU returns
    - SW1 SW2 == 6A 86.

10. EXTERNAL AUTHENTICATE

CLA	00
INS	82
P1	algorithm identifier (4 bits)•security level (4 bits)
P2	00 for default key; 01 to 30 for key number
L_c	length of the cryptogram
Data Field	cryptogram
L_e	empty

References:

1. GSC-IS 5.3.5.2.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.
2. External authentication has not been established.

Assertion 10.1

Purpose: To test EXTERNAL AUTHENTICATE using valid parameters.

Scenario:

1. P1 == a valid algorithm identifier•a valid security level.
2. P2 == a valid key number.
3. L_c == the length of the cryptogram in the data field.
4. Data field == a valid encrypted challenge.
5. An EXTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00
   - those applets on the selected card subject to external authentication are opened to appropriate access.

Assertion 10.2

Purpose: To test EXTERNAL AUTHENTICATE using a bad cryptogram.

Scenario:

1. P1 == a valid algorithm identifier•a valid security level.
2. P2 == a valid key number.
3. L_c == the length of the cryptogram in the data field.

4. Data field == an invalid encrypted challenge.

5. An EXTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 63 CX or 69 83
   - those applets on the selected card subject to external authentication are not opened to appropriate access.

Assertion 10.3

Purpose: To test EXTERNAL AUTHENTICATE using a bad data field length.

Scenario:

1. P1 == a valid algorithm identifier•a valid security level.

2. P2 == a valid key number.

3. L_c /= the length of the cryptogram in the data field.

4. Data field == a valid encrypted challenge.

5. An EXTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00
   - those applets on the selected card subject to external authentication are not opened to appropriate access.

Assertion 10.4

Purpose: To test EXTERNAL AUTHENTICATE in a context where the command is not allowed.

Scenario:

1. P1 == a valid algorithm identifier•a valid security level.

2. P2 == a valid key number.

3. L_c == the length of the cryptogram in the data field.

4. Data field == a valid encrypted challenge.

5. The immediately preceding command was not GET CHALLENGE.

6. An EXTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 69 85
   - those applets on the selected card subject to external authentication are not opened to appropriate access.

Assertion 10.5

Purpose: To test EXTERNAL AUTHENTICATE using invalid parameters P1-P2.

Scenario:

1. At least one of the following is true:
   - P1 /= a valid algorithm identifier•a valid security level.
   - P2 /= a valid key number.

2. L_c == the length of the cryptogram in the data field.

3. Data field == a valid encrypted challenge.

4. An EXTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86 or 6A 88
   - those applets on the selected card subject to external authentication are not opened to appropriate access.

11. INTERNAL AUTHENTICATE

CLA	00
INS	88
P1	00 for default DES3-ECB or Algorithm ID
P2	00 for default key; 01 to 30 for key number
L_c	length of the challenge
Data Field	challenge
L_e	maximum size of encrypted challenge

References:
1. GSC-IS 5.3.5.3.

Starting State for Each Assertion:
1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 11.1

Purpose: To test INTERNAL AUTHENTICATE using valid parameters.

Scenario:
1. P1 == either 00 or a valid algorithm ID.
2. P2 == a valid key number.
3. L_c == the length of the challenge in the data field.
4. Data field == a valid challenge.
5. L_e >= the length of the response.
6. An INTERNAL AUTHENTICATE APDU is issued.

Expected Results:
1. The APDU returns
   - SW1 SW2 == 90 00
   - Response == the returned encrypted challenge.

Assertion 11.2

Purpose: To test INTERNAL AUTHENTICATE using a bad data field length.

Scenario:
1. P1 == either 00 or a valid algorithm ID.
2. P2 == a valid key number.
3. L_c /= the length of the challenge in the data field.
4. Data field == a valid challenge.

5. L_e >= the length of the response.

6. An INTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 11.3

Purpose: To test INTERNAL AUTHENTICATE where the specified length of the returned challenge is incorrect.

Scenario:

1. P1 == either 00 or a valid algorithm ID.

2. P2 == a valid key number.

3. L_c == the length of the challenge in the data field.

4. Data field == a valid challenge.

5. L_e < the length of the response.

6. An INTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6C XX.

Assertion 11.4

Purpose: To test INTERNAL AUTHENTICATE using invalid parameters P1-P2.

Scenario:

1. At least one of the following is true:
   - P1 /= either 00 or a valid algorithm ID.
   - P2 /= a valid key number.

2. L_c == the length of the challenge in the data field.

3. Data field == a valid challenge.

4. L_e >= the length of the response.

5. An INTERNAL AUTHENTICATE APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86 or 6A 88.

12. PRIVATE SIGN/DECRYPT

CLA	80
INS	42
P1	00
P2	00
L_c	length of the data to sign or decrypt
Data Field	data to sign or decrypt
L_e	expected length of the signature/decryption

References:

1. GSC-IS 5.3.6.1.

Starting State for Each Assertion:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Assertion 12.1

Purpose: To test PRIVATE SIGN/DECRYPT using valid parameters.

Scenario:

1. L_c == the length of the data to sign or decrypt in the data field.

2. Data field == data to sign or decrypt.

3. L_e == the expected length of the signature/decryption response.

4. A PRIVATE SIGN/DECRYPT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 90 00
   - Response == the returned signature/decryption.

Assertion 12.2

Purpose: To test PRIVATE SIGN/DECRYPT using a bad data field length.

Scenario:

1. L_c /= the length of the data to sign or decrypt in the data field.

2. Data field == data to sign or decrypt.

3. L_e == the expected length of the signature/decryption response.

4. A PRIVATE SIGN/DECRYPT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 67 00.

Assertion 12.3

Purpose: To test PRIVATE SIGN/DECRYPT where the specified length of the returned signature/decryption is incorrect.

Scenario:

1. L_c == the length of the data to sign or decrypt in the data field.

2. Data field == data to sign or decrypt.

3. L_e <= the length of the signature/decryption response.

4. A PRIVATE SIGN/DECRYPT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6C XX.

Assertion 12.4

Purpose: To test PRIVATE SIGN/DECRYPT using invalid parameters P1-P2.

Scenario:

1. P1 /= 00 and/or P2 /= 00.

2. L_c == the length of the data to sign or decrypt in the data field.

3. Data field == data to sign or decrypt.

4. L_e == the length of the signature/decryption response.

5. A PRIVATE SIGN/DECRYPT APDU is issued.

Expected Results:

1. The APDU returns
   - SW1 SW2 == 6A 86 or 6A 88.